Sobre as Operações de Redução do SCP

LÍDIO NUNES DE ABREU
HUMBERTO JOSÉ LONGO
Instituto de Informática - UFG
Caixa Postal 131, Goiânia - GO, Brasil
CEP: 74001-970

Resumo

Palavras chaves:

Abstract

Keywords:

I. Recobrimento de um Conjunto

O recobrimento de um conjunto, considerando-se a existência de um número limitado de subconjuntos do mesmo, é equivalente àqueles que unidos geram tal conjunto. Formalmente, dados os conjuntos $J=\{1,...,n\}$, $I=\{1,...,m\}$ e uma família $F=\{I_1,...,I_n\}$ de subconjuntos de I ($I_j \subseteq I$, $\forall\ j \in J$), então qualquer subconjunto $J^* \subseteq J$ define um recobrimento do conjunto I, se $\bigcup_{j\in J^*} I_j = I$. Associando-se a cada subconjunto j da família F um custo c_j, então um recobrimento J^* terá um custo total de $\sum_{j\in J^*} c_j$. A minimização deste custo é chamado de Problema de Recobrimento de um Conjunto (*SCP – Set Covering Problem*).

O *SCP* é um problema NP-difícil (Karp, 1972) que vem sendo exaustivamente estudado. Este fato pode ser atribuído ao grande número de aplicações que podem ser formuladas como *SCP*. Deste modo, encontram-se na literatura diversos métodos aproximados e exatos para sua resolução. Vários desses métodos utilizam-se de uma fase de pré-processamento que, dependendo das características de uma dada instância, procura eliminar da mesma elementos e subconjuntos, sem que se incorra na quebra das restrições estabelecidas ou, ainda, identificar subconjuntos que possam ser previamente incluídos na solução da mesma..

Este trabalho, além de listar algumas regras básicas para testar a factibilidade ou reduzir o tamanho de uma determinada instância do SCP, mostra a complexidade da execução de cada uma dessas regras e como os algoritmos mais conhecidos para a resolução do *SCP* as utilizam. As referências básicas para tais regras são Lemke, Salkim e Spielberg, 1971 e Fisher e Kedia, 1990.

II. Propriedades e Regras de Redução do *SCP*

Além dos conjuntos já definidos na seção anterior, será usado nesta seção, na descrição das propriedades e regras a seguir, o conjunto $J_i = \{j \in J \mid i \in I_j\}$, $\forall\ i \in I$, que corresponde aos subconjuntos da instância em questão que contêm o elemento i. Uma convenção adotada é a de que, dados dois subconjuntos Ii e Ij da família F, se $i \leq j$, então $c_i \leq c_j$.

1. **Custos positivos :** pode-se assumir que $c_j > 0$, $\forall\ j \in J$. Se $c_j \leq 0$, para algum $j \in J$, então pode-se inserir o subconjunto I_j na solução e exclui-lo da instância. Consequentemente, todo elemento

$i \in I_j$ também pode ser excluído (uma vez que já estarão recobertos por I_j). Após este processo pode ocorrer que, para algum $j' \in J$, a cardinalidade de $|I_{j'}|$ torne-se zero e, assim, pode-se excluir o subconjunto $I_{j'}$ também.

2. **Factibilidade :** a instância é infactível se $|J_i| = 0$ para algum $i \in I$. Claramente, neste caso, o elemento i não pertence a nenhum dos subconjuntos I_j, $j \in J$, e não pode ser recoberto.

3. **Subconjuntos pré-fixados :** se $|J_i| = 1$ para algum $i \in I$, pode-se inserir na solução o subconjunto I_j, para o único $j \in J_i$. A conseqüência dessa propriedade é que pode-se excluir todo elemento $k \in I_j$, pois os mesmos já estarão recobertos pelo subconjunto I_j.

4. **Dominância de elemento :** se para quaisquer i_1, $i_2 \in I$ tem-se que $J_{i_1} \subseteq J_{i_2}$, então pode-se excluir da instância todo subconjunto $j \in (J_{i_1} \setminus J_{i_2})$ tal que $j \notin J_{i_3}$ para algum $i_3 \in I$ e $i_3 \neq i_1$, pois tais subconjuntos não são necessários para recobrir o elemento i_2 e nenhum outro mais. Além do elemento i_2, o elemento i_1 também pode ser excluído, uma vez que ele será recoberto pelo mesmo subconjunto que recobriu i_2.

5. **Dominância de subconjuntos :** se para algum $J' \subset J$ e $k \in J \setminus J'$ tem-se que $I_k \subseteq \bigcup_{j\in J'} I_j$ e $\sum_{j\in J'} c_j < c_k$, então o subconjunto I_k pode ser excluído, pois o custo de incluí-lo na solução é no mínimo igual ao da inclusão de todos os subconjuntos $j \in J'$.

6. **Subconjunto de custo dominado :** para todo $i \in I$, seja $d_i = (\min c_j \mid j \in J_i)$. Se para algum $j \in J$ tem-se que $c_j > \sum_{i\in Ij} d_i$, então o subconjunto j pode ser excluído, pois os elementos por ele recobertos podem sê-los com menor custo por outros subconjuntos.

III. Complexidade das regras de redução

O cálculo da complexidade das regras de redução 1 a 3 é trivial, sendo $O(n)$ para a primeira e $O(m)$ para as outras duas. Assim, nos limitaremos aqui a analisar as regras 4 a 6.

Na regra de redução número 5 (Dominância de subconjuntos), qualquer subconjunto I_j, $j \in J$, pode ser excluído da instância, caso haja uma composição (união) de subconjuntos pertencentes à família F, tal que todo elemento de I_j pertença a pelo menos um subconjunto da composição e o custo total desta seja inferior ao custo de I_j.

Na avaliação da possibilidade de exclusão de um subconjunto I_j, $j \in J$, usando-se essa regra, qualquer composição de subconjuntos, cujo custo resultante não seja superior ao custo de c_j, é válida.

Assim, o número total de composições factíveis (λ) para a exclusão de um determinado subconjunto I_j da instância, considerando-se que $c_j > 0$, $\forall\ j \in$ J, é a soma de todas as combinações de tamanho δ, onde $\delta \in \{1, ..., j-1\}$, ou seja, $\lambda = \binom{j-1}{1} + \ldots + \binom{j-1}{j-1} = \sum_{\delta=1}^{j-1} \binom{j-1}{\delta}$. Como $\sum_{i=0}^{n} \binom{n}{i} = \binom{n}{0} + \sum_{i=1}^{n} \binom{n}{i} = 2^n$, então $\lambda = 2^{j-1} - 1$.

Portanto, concluí-se que apenas algoritmos de ordem exponencial, em relação ao número de subconjuntos de custo menor ao daquele a ser excluído, podem ser obtidos para esta redução.

IV. Alternativas às reduções 5 e 6

Para a exclusão de um subconjunto I_j, $j \in J$, qualquer composição de subconjuntos, cujo custo total seja não superior ao de I_j, é uma possível alternativa onde pode-se verificar as características mencionadas.

Considerando-se que existem duas restrições a serem satisfeitas pelas composições (custos e a pertinência dos elementos de I_j, então o número de alternativas que satisfazem a ambas é provavelmente muito inferior ao número de possibilidades rejeitadas (o que compreende as alternativas que não satisfazem a uma ou ambas restrições). Pode-se explorar estas características aplicando-se métodos analíticos sobre os dados, possibilitando-se de alguma forma dividi-los sobre um dos aspectos (custo/pertinência) e realizando-se a partir das informações obtidas a exclusão de parte das alternativas.

A metodologia empregada deve ser fácil e eficiente, de forma a não influir negativamente na ordem de complexidade da redução. Um procedimento relativamente eficiente que pode ser feito é agrupar os subconjuntos precedentes a I_j de tal forma a obter 2 regiões, sendo que a primeira delas terá um custo total não superior ao custo de I_j. Denotando-se por k, $k \in \{1, ..., j-1\}$ o subconjunto de maior custo da primeira região, temos que as regiões ficam assim divididas:

- $I_1, ..., I_k$ - região com custo total não superior a c_j;
- $I_{k+1}, ..., I_j$ - subconjuntos restantes.

Para qualquer elemento k uma possível composição que pode possibilitar a exclusão do subconjunto I_j é a própria região $I_1, ..., I_k$. Quando esta região não satisfaz os critérios impostos, então nenhuma de suas partições satisfará, logo 2^k-2 alternativas são descartadas. As outras alternativas possíveis são as composições resultantes da substituição de alguns subconjuntos por outros pertencentes a $\{I_{k+1}, ..., I_{j-1}\}$ de modo que o custo resultante permaneça não superior a c_j. Considerando-se que o custo de cada um destes subconjuntos é no mínimo igual a c_k, então a quantidade de subconjuntos inseridos para formar uma nova composição, não pode ser superior a quantidade de subconjuntos retirados. As partições de $\{I_{k+1}, ..., I_{j-1}\}$ com cardinalidade não superior a k podem eventualmente possuir um custo resultante inferior ao custo de I_j, e portanto também constituem possíveis composições. Independentemente do resultado da aplicação deste procedimento analítico, os subconjuntos $\{I_{k+1}, ..., I_n\}$ individualmente também constituem composições factíveis.

Para uma dada instância o número de alternativas é a união de todas as possibilidades presentes nestes casos. Quando o valor de k é $j-1$ as possibilidades a serem verificadas são o próprio conjunto $\{I_1, ..., I_k\}$ e os subconjuntos posteriores a I_j, fazendo com que o número de composições factíveis seja $1+n-j$. Para os outros valores do domínio de k o número de possibilidades pode ser definido pela seguinte expressão: $1 + \sum_{q=1}^{\omega} C_q^{n-k-1} + n - j$, onde $k \in \{1, ..., j-2\}$ e $\omega = min(n-k-1, k-p)$.

A similaridade existente entre as reduções Dominância de Conjuntos e Conjunto de Custo Dominado pode ser verificadas em diversos casos, entretanto um número não desprezível de exclusões são particulares à primeira. Isto ocorre porque a redução Conjunto de Custo Dominado trabalha com uma quantidade reduzida de subconjuntos, avaliando a existência de possíveis composições destes que permita excluir um subconjunto I_j qualquer. Os subconjuntos citados referem-se aos de menor custo que contém o elemento $i \in I$. Como conseqüência, esta redução não prevê os casos onde um subconjunto de custo mais elevado pode integrar uma composição que permita eliminar um subconjunto I_j. A unicidade dos subconjuntos nestas composições também não é imposta. Contudo, mesmo restringindo-se a análise a poucos subconjuntos, apenas uma parcela das possíveis combinações dos mesmos é avaliada. Este é o fator limitante da abrangência desta redução, quando comparada com a redução Dominância de Conjuntos, o que explica a complexidade de ordem polinomial associada a esta redução.

V. Redução 'Conjunto de Custo Dominado' modificada

Se para algum $\lambda \in J$ tem-se que $c_j > c_{M(Tj)}$, onde $T_j = \cup_{i \in P_j} d_i$ e $M(T_j)$ corresponde a composição ótima dos elementos de T_j, ou seja, a composição de mínimo custo que satisfaz todas as restrições de P_j, então o subconjunto P_λ pode ser eliminado pois os elementos por ele recobertos podem sê-los com menor custo por T_j.

Esta reformulação na definição desta redução busca abranger uma quantidade maior de exclusões, mantendo-se a ordem de complexidade polinomial. Infelizmente, determinar esta composição trata-se de uma sub-instância de recobrimento, onde o conjunto I corresponde aos elementos de P_j e a família P os subconjuntos de T_j. Considerando a complexidade de problemas como o SCP, determinar a composição de custo mínimo envolverá uma ordem de complexidade exponencial, e mesmo neste caso esta redução ainda não possuirá a abrangência da quarta redução.

A complexidade desta redução esta no fato de se determinar $M(T_j)$ e não um dos possíveis T_j. Decorre que se relaxarmos um pouco a definição, pode-se voltar a ter uma redução com complexidade polinomial que apesar de ser mais fraca que a definição proposta ainda assim consegue ser mais abrangente que a definição original.

VI. Redução proposta

Para $\forall i \in I$, seja $d_i = \{j \in J \mid i \in P_j$ \wedge c_j é mínimo$\}$. Definindo-se $T_j = \cup_{i \in P_j} d_i$ então se para algum $\lambda \in J$ tem-se que $c_\lambda > \sum_{j \in T_j} c_j$, então o subconjunto P_λ pode ser eliminado pois os elementos por ele recobertos podem sê-los com menor custo por T_j.

Esta nova definição tem a vantagem de ser mais rigorosa quanto a formação das composições avaliadas, e não desconsidera as situações avaliadas pela definição original. A principal dificuldade para abranger a redução coluna de custo dominado há todas as exclusões obtidas pela redução dominância de coluna, é o fato de que o objetivo não é garantir que cada elemento $i \in I$ esteja associado no recobrimento com o subconjunto de menor custo que o contém, mas garantir que todo elemento i está associado a algum subconjunto P_j do recobrimento J^* e o custo deste c_{j^{*} } é mínimo.

Referências

[1] - Fisher, M. L. e P. Kedia, *Optimal Solution of Set Covering/Partitioning Problems using Dual Heuristics*, Management Science 36(6):674-688, Junho 1990.

[2] - Karp, R. M., *Complexity of Computer Computations, Reducibility Among Combinatorial Problems*, 85-104, Plenum Press, 1972.

[3] - Lemke, C. e H. Salkin e K. Spielber , *Set Covering by Single Branch Enumeration*, Operations Research 4:998-1022, April 1971

Contudo, dentre estes, mesmo os de melhor desempenho não conseguem resolver satisfatoriamente determinadas classes de instâncias. Seja por não se adequarem à instância (em especial no caso aproximado), seja pela dificuldade de se identificar que a instância corresponde a um caso particular.

a) os subconjuntos distintos de I_j e que tenham custo igual ao deste não podem participar de composições com os demais subconjuntos;
b) com relação aos subconjuntos I_k, $k \in \{1, ..., j\text{-}1\}$, supondo $c_k < c_j$, a presença destes não pode ser restringida a composições individuais sem antes conhecer exatamente quais são estes subconjuntos;
c) em relação aos subconjuntos I_k, $k \in \{j+1, ..., n\}$ e $c_k = c_j$, pode-se restringir as possíveis composições com as mesmas e o número de alternativas presentes neste caso; e
d) os subconjuntos com custos superiores ao custo de I_j, caso existam, não estarão presentes em nenhuma composição que possibilite excluir o subconjunto I_j.

Balas, E. and A. Ho, *Set Covering Algorithms using Cutting Planes, Heuristics, and Subgradient Optimization: A Computational Study*, Mathematical Programming Study 12:37-60, April 1980.

Beasley, J.E., *An Algorithm for Set Covering Problem*, European Journal of Operational Research, 31(1):85-93, Julho 1987.

Na análise a ser realizada, a definição da redução será estendida, permitindo-se composições com o mesmo custo do subconjunto I_j a ser excluído. Assim, um limite inferior para o desempenho dos algoritmos para a quarta redução pode ser obtido, determinando-se o número mínimo de avaliações a serem efetuadas, para se verificar a existência de tais composições.

Na construção de uma composição para a exclusão de um subconjunto I_j de uma família F, os subconjuntos podem ser divididos três grupos de acordo com os custos:
e) G1 = $\{I_k \mid k \in \{1, ..., j\text{-}1\}$ e $c_k = c_j\}$;
f) G2 = $\{I_k, \mid k \in \{1, ..., j\text{-}1\}$ e $c_k < c_j\}$; e
g) G3 = $\{I_k, k \in \{j+1, ..., n\}$ e $c_k > c_j\}$.